Practical-3

Retrieving, Restricting and Sorting Data (DRL)

```
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    hire_date DATE,
    salary DECIMAL(10, 2),
    department VARCHAR(50),
    job_title VARCHAR(50),
    manager_id INT,
    status VARCHAR(20)
);
```

DRL Queries

Here are 20 different queries for retrieving, restricting, and sorting data:

1. **Retrieve all columns for all employees**:
   ```sql
   SELECT * FROM employees;
   ```

2. **Retrieve only the first_name and last_name of all employees**:
   ```sql
   SELECT first_name, last_name FROM employees;
   ```

3. **Retrieve employees with a salary greater than 50,000**:
   ```sql
   SELECT * FROM employees WHERE salary > 50000;
   ```

4. **Retrieve employees who were hired after January 1, 2020**:
   ```sql
   ```

```
SELECT * FROM employees WHERE hire_date > '2020-01-01';
```

5. **Retrieve employees in the 'Sales' department**:
```sql
SELECT * FROM employees WHERE department = 'Sales';
```

6. **Retrieve employees whose last name is 'Smith' and order by hire_date**:
```sql
SELECT * FROM employees WHERE last_name = 'Smith' ORDER BY hire_date;
```

7. **Retrieve employees with salaries between 40,000 and 60,000, ordered by salary descending**:
```sql
SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000 ORDER BY salary DESC;
```

8. **Retrieve employees who do not have a manager (manager_id is NULL)**:
```sql
SELECT * FROM employees WHERE manager_id IS NULL;
```

9. **Retrieve the top 5 highest-paid employees**:
```sql
SELECT * FROM employees ORDER BY salary DESC LIMIT 5;
```

10. **Retrieve employees with 'Manager' in their job title**:
```sql
SELECT * FROM employees WHERE job_title LIKE '%Manager %';
```

11. **Retrieve the count of employees in each department**:
    ```sql
    SELECT department, COUNT(*) AS employee_count FROM employees GROUP BY department;
    ```

12. **Retrieve the average salary of employees in the 'Engineering' department**:
    ```sql
    SELECT AVG(salary) AS average_salary FROM employees WHERE department = 'Engineering';
    ```

13. **Retrieve the maximum salary among all employees**:
    ```sql
    SELECT MAX(salary) AS max_salary FROM employees;
    ```

14. **Retrieve employees ordered by last_name and first_name**:
    ```sql
    SELECT * FROM employees ORDER BY last_name, first_name;
    ```

15. **Retrieve employees whose first_name starts with 'J'**:
    ```sql
    SELECT * FROM employees WHERE first_name LIKE 'J%';
    ```

16. **Retrieve employees with a salary lower than the average salary**:
    ```sql
    SELECT * FROM employees WHERE salary < (SELECT AVG(salary) FROM employees);
    ```

17. **Retrieve employees who were hired in 2023**:
    ```sql
    SELECT * FROM employees WHERE YEAR(hire_date) = 2023;
    ```

18. **Retrieve employees who have a status of 'Active' and sort by email**:
    ```sql
    SELECT * FROM employees WHERE status = 'Active' ORDER BY email;
    ```

19. **Retrieve employees with salaries greater than the average salary of their department**:
    ```sql
    SELECT e1.*
    FROM employees e1
    JOIN (
        SELECT department, AVG(salary) AS avg_salary
        FROM employees
        GROUP BY department
    ) e2 ON e1.department = e2.department
    WHERE e1.salary > e2.avg_salary;
    ```

20. **Retrieve distinct job titles in the company**:
    ```sql
    SELECT DISTINCT job_title FROM employees;
    ```

Basic Retrieval

1. **Retrieve all columns for all employees**:
   ```sql
   SELECT * FROM employees;
   ```

2. **Retrieve only the `first_name` and `last_name` of all employees**:
   ```sql
   SELECT first_name, last_name FROM employees;
   ```

Filtering Data

3. **Retrieve employees with a salary greater than 50,000**:
   ```sql
   SELECT * FROM employees WHERE salary > 50000;
   ```

4. **Retrieve employees hired after January 1, 2020**:
   ```sql
   SELECT * FROM employees WHERE hire_date > '2020-01-01';
   ```

5. **Retrieve employees in the 'Sales' department**:
   ```sql
   SELECT * FROM employees WHERE department = 'Sales';
   ```

6. **Retrieve employees whose `last_name` is 'Smith'**:
   ```sql
   SELECT * FROM employees WHERE last_name = 'Smith';
   ```

7. **Retrieve employees whose `first_name` starts with 'A'**:
   ```sql
   SELECT * FROM employees WHERE first_name LIKE 'A%';
   ```

8. **Retrieve employees whose `status` is 'Active'**:
   ```sql
   SELECT * FROM employees WHERE status = 'Active';
   ```

9. **Retrieve employees with a salary between 40,000 and 60,000**:
   ```sql
    SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000;
   ```

10. **Retrieve employees with salaries less than 30,000 or greater than 70,000**:
    ```sql
    SELECT * FROM employees WHERE salary < 30000 OR salary > 70000;
    ```

11. **Retrieve employees who do not have a manager (`manager_id` is NULL)**:
    ```sql
    SELECT * FROM employees WHERE manager_id IS NULL;
    ```

12. **Retrieve employees who have 'Engineer' in their job title**:
    ```sql
        SELECT * FROM employees WHERE job_title LIKE '%Engineer%';
    ```

13. **Retrieve employees who were hired on January 15, 2023**:
    ```sql
    SELECT * FROM employees WHERE hire_date = '2023-01-15';
    ```

Sorting Data

14. **Retrieve employees ordered by `hire_date`**:
    ```sql
    SELECT * FROM employees ORDER BY hire_date;
    ```

15. **Retrieve employees ordered by `salary` in descending order**:
    ```sql
    SELECT * FROM employees ORDER BY salary DESC;
    ```

16. **Retrieve the top 10 highest-paid employees**:
    ```sql
    ```

```
SELECT * FROM employees ORDER BY salary DESC LIMIT 10;
```

17. **Retrieve employees ordered by `department` and then by `last_name`**:
```sql
SELECT * FROM employees ORDER BY department, last_name;
```

18. **Retrieve employees with the lowest 5 salaries**:
```sql
SELECT * FROM employees ORDER BY salary ASC LIMIT 5;
```

Aggregation

19. **Retrieve the total number of employees**:
```sql
SELECT COUNT(*) AS total_employees FROM employees;
```

20. **Retrieve the average salary of all employees**:
```sql
SELECT AVG(salary) AS average_salary FROM employees;
```

21. **Retrieve the highest salary**:
```sql
SELECT MAX(salary) AS max_salary FROM employees;
```

22. **Retrieve the lowest salary**:
```sql
SELECT MIN(salary) AS min_salary FROM employees;
```

23. **Retrieve the number of employees in each department**:

```sql
SELECT department, COUNT(*) AS employee_count FROM employees GROUP BY department;
```

24. **Retrieve the total salary expenditure**:
```sql
SELECT SUM(salary) AS total_salary FROM employees;
```

25. **Retrieve the average salary for employees with a specific job title**:
```sql
SELECT AVG(salary) AS avg_salary FROM employees WHERE job_title = 'Senior Developer';
```

Advanced Queries

26. **Retrieve employees whose salary is above the average salary for their department**:
```sql
SELECT e1.*
FROM employees e1
JOIN (
    SELECT department, AVG(salary) AS avg_salary
    FROM employees
    GROUP BY department
) e2 ON e1.department = e2.department
WHERE e1.salary > e2.avg_salary;
```

27. **Retrieve employees who have been with the company for more than 5 years**:
```sql
SELECT * FROM employees WHERE hire_date < DATE_SUB(CURDATE(), INTERVAL 5 YEAR);
```

28. **Retrieve employees who report to a manager in the 'Engineering' department**:
    ```sql
    SELECT e1.*
    FROM employees e1
    JOIN employees e2 ON e1.manager_id = e2.employee_id
    WHERE e2.department = 'Engineering';
    ```

29. **Retrieve the distinct job titles of employees**:
    ```sql
    SELECT DISTINCT job_title FROM employees;
    ```

30. **Retrieve employees with a salary within the top 25% of salaries**:
    ```sql
    SELECT * FROM employees
        WHERE salary >= (SELECT PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY salary) FROM employees);
    ```